DOCUMENT RESUME

ED 122 830 . IR 003 448

AUTHOR Hedberg, John G.
TITLE The Technologies of EXPER SIM.
PUB DATE 21 Apr 76
NOTE 26p.; Paper presented at the Annual Meeting of the American Educational Research Association (San Francisco, California, April 19-23, 1976); For related documents see IR 003 446 and 451; Not available in hard copy due to print quality of original document

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS College Instruction; *Computer Assisted Instruction; *Computer Programs; Higher Education; Instructional Technology; Models; On Line Systems; Research Methodology; *Simulation; Statistical Analysis; Tables (Data); Teaching Techniques
IDENTIFIERS *EXPER SIM; Interactive Computer Programs; LESS; MESS; Simulation Models

ABSTRACT

EXPER SIM has been translated into two basic software systems: the Michigan Experimental Simulation Supervisor (MESS) and Louisville Experiment Simulation Supervisor (LESS). MESS and LESS have been programed to facilitate student interaction with the computer for research purposes. The programs contain models for several statistical analyses, and new models can be added to these simulation systems. Models available cover a wide spectrum of disciplines in the physical science as well as the social sciences. The system is flexible and the package of programs involves a supervisor that controls the student's interaction. It also involves a series of related data-generating subroutines which are used by each model in the development of the data generating algorithm. (CH)

THE TECHNOLOGIES OF EXPER SIM

John G. Hedberg
Syracuse University

A paper presented at the American Educational Research Association. Annual Meeting,

San Francisco, California.
April 21, 1976.

Section 15.19 Experimental research simulation: The Pedagogy, Technologies, Applications and Evaluation of EXPER SIM.

The technologies of EXPER SIM.

John G. Hedberg

Syracuse University

EXPER SIM as a pedagogy has been translated into two basic software systems. The larger system commonly called MESS for 'Michigan Experimental Simulation Supervisor' was originally designed and implemented at the University of Michigan under the guidance of Dana Main. The MESS programs began as a series of subroutines that were combined to provide a flexible system. MESS was written in Fortran and designed to be an interactive system, responding to student input about the experimental simulation.

The smaller system was written at the University of Louisville and has been dubbed by its authors, Art Cromer and John Thurmond, LESS (Louisville Experiment Simulation Supervisor). The LESS system, written in the BASIC language, was designed for a much smaller machine and, since it was a later development to MESS, it includes additional features such as a simulation writing program that creates new simulations in a fraction of the time required by the MESS system. Some differences between these two systems are summarized in Table 1.

The software from a student's perspective

The potential of the EXPER SIM approach may be best illustrated by examining possible student interactions and experimental runs. The major software difference between MESS and LESS is the way in which the simulated experimental results are calculated. Consider, for example, a single 2 by 2 factorial design. LESS requires a separate run to generate the data for each cell in the design, whereas MESS can generate the data for all cells in the design in a single run. Using MESS the student can enter different levels of each variable at the one time and all combinations will be calculated and outputted together.

LESS Simulations. In the LESS system the student can easily specify and run any of the available models. The interaction may be ~~seen as~~ consisting of four components; the selection of a simulation model, the specification of each variable's value or level, the data output and summary statistics. An example of a student interaction using the Fear simulation is given in Figure 1.

Each model is acessed by responding to the prompt of the supervisor for example, Fear is selected from a number of alternatives in Figure 1. With this LESS simulation three categories of input variables must be specified in a predetermined order; number of subjects, dependent measures, and independent variables. The program is written to provide prompting should a student forget the correct order. The number of subjects to be run is the first input variable (or general parameter) to be entered. Then since the simulation has a number of dependent measures, a variable which determines which will be outputted is entered next. For example, in Figure 1, MEASURE is such a variable; it has three valid levels 1,2,3 where 1 generates only the EXPLORE dependent measure, 2 generates only the THRUST dependent measure, and 3 generates both EXPLORE and THRUST measures. Last, an input variable for each independent variable to be manipulated by the students must be entered. In the FEAR simulation there are two independent variables: FEAR, which is a discrete variable with 5 levels and TESTOST, which is continuous variable with a range of 0 to 100.

For each condition the values of the input variables must be specified. In Figure 1, one condition is run with 15 subjects, measuring both dependent variables and using FEAR at level 2, with TESTOST at 87. Once each input variable has been specified the LESS program generates data for the required number of cases and (if the model provides for it) produces some summary statistics. The student has now generated data under one set of input variable values or one condition; the program will then ask for further conditions to be specified.

MESS Simulations. The student interacts with MESS by using a number of commands which form a student-operated management system. Whereas LESS requests the values of the variables immediately when the particular model is specified, MESS asks the student to enter a command that might, for example, run an experiment, stop an experiment, or change the amount of output received at the terminal. Thus MESS, unlike LESS does not immediately set up the first condition. The interaction thus begins (once a student has signed on) by the student giving a command ">>EXPT", whereupon the computer requests information analogous to the input variable values of LESS. An example of a MESS interaction is given in Figure 2.

The supervisor program has a few additional features to the LESS simulations. The first line of information requested by the program is a general identification line which will be printed as a heading on the output. The second request is for the number of conditions that will be defined. This is a major difference from LESS which deals with only one condition at a time. MESS will expect each of the specified number of conditions to be defined before the output is printed or calculated for each condition. The third line defines the experimental conditions in terms of variable levels and values.

The MESS models can be constructed to allow a range of alternative spellings or abbreviations for each variable name. Rather than being prompted on each variable or specifying them in a particular sequence, MESS allows a combination of variables at multiple levels to be defined in the one line. For example, in Figure 2, (section d), three variables are given particular values. MESS's flexibility allows a large number of conditions to be defined with minimal effort. Further, the variables can exist in a number of formats, they can be specified by nominal categories, by integer values (like LESS), or by combinations of these two, and by decimal values (for continuous variables). Some logical variables are also available to enable the selection of dependent measures or the printing of randomly fluctuating variables which are not being controlled. Any variable that is not specifically mentioned in the definition line(s) will assume a default value if it has been prespecified in the model.

Once the conditions have been defined the MESS program prints the values for each variable under each condition. It is only at this point that the number of subjects to be run is requested (Figure 2, section i). To enable different numbers of subjects to be specified for each condition (i.e. to allow unequal cell sizes) MESS allows a series of values to be entered. Since the conditions have been defined and have unique labels associated with them (A, B, C, D in Figure 2, section 1), MESS will even allow the student at this point to specify a repeated measures design should that be allowed in the model. In Figure 2 with an imprinting model this is not available and a cell size of 15 is entered. The output from the experiment is then typed at the terminal, with any summary statistics that might be produced by the model.

The software from an instructor's perspective

Perhaps the major questions facing anyone who intends to use the LESS and MESS systems are three:

1. What range of models are currently available?

2. What can be changed with existing models?

3. How easily can new models be developed?

To answer each of these questions in turn, it is important to recognize the different structures of the two systems. LESS is a smaller and much more closely integrated program (Figure 3). The models that are currently programmed for LESS cover a wide variety of psychological areas and require a number of statistical methods for the analysis of the data produced. In their instructor's manual Thurmond and Cromer (1974) discuss each model in terms of its potential for a certain type of statistical analysis: alternatives include correlation, t-test, ANOVA, and Chi-square.

In the construction of LESS, the importance of easily generated models has been recognized by the authors and has been built into the LESS system. In figure 3, two approaches to model development are illustrated, in one the author is required to write a program (author mode) which is then chained to the main LESS program at the time of execution. In the second, a separate program is not required; rather a series of variables are entered into a regression model and the weights and effects are manipulated to achieve the desired simulation performance. This second alternative has been dubbed the General Model Builder (GMB) and is a unique feature of LESS. There are constraints such as the type of model specified and the number of variables allowed. Using this general model builder it is possible to accommodate a maximum of 20 variables. The GMB will allow up to 10 dependent variables with a total of approximately 350 terms and effects. The types of variables available to the simulation designer in LESS are:

1. Parameter variables which determine the number of dependent variables that are generated in one run. Examples are number of subjects, cases, time periods etc.

2. Discrete variables which can take integer values only and generally signify levels of a variable, e.g. Sex, 1=male, 2=female. There may be a maximum of 25 levels of a discrete variable.

3. Continuous variables which may be used in an author written program directly or may be discretized for use with the GMB.

4. "X" variables which are unknown variables and which may represent one or more specific variables (i.e. Secondary) to be discovered by the student, or which the author did not include in the original model.

5. Secondary variables which are the specific variables not revealed to the student initially but are important to the model.

6. Dependent variables which represent the results of measurement or observation in the simulation.

In LESS the effects of these variables are stored in vectors and matricies and they become operable when the student calls for a particular model. The LESSO program is chained to the supervisor for GMB produced simulations and the particular program (LESS1, LESS2, etc.) is chained in the case of author-written programs.

Model building in MESS. The MESS system also has a series of components which are linked together for the execution of model programs. The precise nature of the linkage depends upon which of three "versions" of MESS is employed by the instructor. The versions differ in the type of input data accessed by the supervisor program.

1. Version 3-P is the programmers version; it can check the input and provide useful diagnostics for model development. This version may also be used to run student simulations.

2. Version 3-SB is a student version designed to be more efficient with computer time and use data which has been transposed into binary form.

3. Version 3-SC is a student version designed for card-oriented computer installations.

The basic principles of the system can be illustrated with reference to one of these versions, thus for the remaining discussion version 3-P will be used.

The basic organization of the components is illustrated in Figure 4. The relationship between components may vary slightly between installations depending upon the organization of the computer's operating system. For example, on an IBM machine the model subroutine is link-edited to the main supervisor and any of the required data generating subroutines to produce one package. Each model is thus a complete unit, this can lead to some inefficiency when large numbers of simulations are stored simulataneously as each unit contains its copy of the supervisor. Each link-edited model simulation accesses two files of data (i.e. data sets). The first file, called DATA, contains the model control variables, their ranges and legal values, together with model identification lines, parameters, and valid commands. In fact with the exception of the algorithm that produces the simulated data the DATA file contains all the variable data required for a student's interaction with MESS. The second file, called MINIT (Model Initialization), contains the seed values of effects and probability values that are used by the model; manipulation of these values, which is be done outside the Fortran program, changes the behavior of the model.

By changing the data in the file called DATA, the instructor can do such things as:

1. change the variables manipulated by the student
2. change the names and abbreviations of a variable
3. change the MESS commands available to the student
4. change the type of statistics printed at the end of the generated data for each condition
5. change valid combinations of variables
6. change the headings outputted for the dependent measures

These changes require little knowledge of the system and no knowledge of programming.

The types of variables available to the programmer in MESS are not as clearly identified as in the LESS system. While the system is not set up to operate with "unknown's" as LESS it does provide the capacity to have hidden variables which can be revealed to a student after some preliminary examination of the original variables. The variable types may be categorized as:

1. Simulation control variables (i.e. LESS parameters). These represent the total number of conditions each subject may undergo (up to 16), the number of subjects in each group, the number of subjects in the total experiment, the default number of subjects per group, and if costs are to be printed.

2. Independent variables which consist of integer number variables (i.e. LESS discrete), floating-point number variables (i.e. LESS continuous), nominal variables (i.e. names used for different categories), either nominal or integer variables (interchangeable), and either nominal or floating-point variables.

3. Logical variables which are entered by the student when defining each condition. Logical variables are used to signal what dependent measures are to be printed. If multiple dependent measures are produced by the model then a student might be required to specify only those relevant to his/her study.

There is no MESS counterpart of the GMB available at present. Each model that is added to the system must be programmed in Fortran as a subroutine and linkedited to the main supervisor program. While this in essence is not difficult it does require a knowledge of the language and the way in which the system stores information about the simulations. At the present time work is underway with a reworked MESS system based upon the assumption that model building should be available as an integral part of the system. Under this new version of MESS (called SWIP) the instructor is able to interactively define the algorithm which forms the basis for the data-generating model, and then can specify the variables that form this model. Under this version of MESS, the variables are defined as either manipulable (i.e. independent), central (i.e. use in calculations to produce dependent measures), or observable (i.e. dependent) variables.

In summary, MESS and LESS provide a selection of usable simulation models. However, the addition of new models to the systems may require more effort on the part of the instructor when programming is required. In a survey of some current users of EXPER SIM, 17 of the 20 responding used MESS. However, only 3 or 4 had developed simulation models specific to their needs. This also might be expected from faculty reaction to workshops held at Syracuse University where faculty indicated they would like a more easily accessable model-builder.

Programming and adaption to an installation

Very few changes have been required in adapting either MESS or LESS to different computer systems. In response to a questionnaire, 12 of 19 respondents claimed to have made minor modifications to the systems to adapt them to their installation. With MESS the modifications involved changing a system subroutine that took the time of day and date from the system clock. With LESS no major problems seem to have been encountered although only a small number of respondents (3 out of 19) had implemented the system.

Only a small number of users have been limited to using the MESS system in batch rather than interactive mode. For these it appears to work satisfactorily. When using MESS in a batch mode the student input is read into the computer as a card deck in an order which follows the normal interactive sequence. At Syracuse University the batch input cards have been replaced with a pseudo-interactive system, written in APL, which generates images of all the cards needed (Nielsen and Hedberg, 1976).

Summary

Both MESS and LESS have been programmed to facilitate student interaction. Each system of programs involves a supervisor that controls the student's interaction and a series of related data generating subroutines which are used by each model in the development of the data generating algorithm. New models can be added to these simulation systems, more easily at present on LESS with its GMB capability than on MESS. Models currently available cover a wide spectrum of disciplines. While the original models developed out of an undergraduate psychology course, models are now available in the physical sciences (e.g. Chemistry and Physics) and in the social sciences (e.g. Education and Sociology). In fact the systems are highly flexible and only limited by the imagination of the instructor who seeks to develop simulations in his/her discipline. The package of programs involves a supervisor that controls the student's interaction and a series of related data generating subroutines which are used by each model in the development of the data generating algorithm.

REFERENCES

Main, D. B. (Ed.). _EXPER SIM on michigan experimental simulation supervisor._ Ann Arbor, Michigan: Xerox, 1974.

Nielsen, T. G., & Hedberg, J. G. _EXPER SIM at syracuse university: Student's and instructor's manuals._ Syracuse, N. Y.: Syracuse University, 1976.

Stout, R. L. _MESS: Instructor's guide for the use of michigan experimental simulation supervisor on MTS._ Ann Arbor, Michigan: University of Michigan, 1973.

Stout, R. L. _Programmer' guide to MESS, versions 3-P, 3-SB, 3-SC._ Ann Arbor, Michigan: University of Michigan, 1974.

Thurmond, J. B., & Cromer, A. D. (Eds.). _LESS guide for_ *student users of the louisville experiment simulation supervisor.* Louisville, Kentucky: University of Louisville, 1974.

Thurmond, J. B., & Cromer, A. D. (Eds.). *LESS instructor's guide for the use of the louisville experiment simulation supervisor.* Louisville, Kentucky: University of Louisville, 1974.

Table 1 Some comparisons between MESS and LESS.

Factor	MESS	LESS
Hardware features.		
Core requirements	150K	20K
Original language	Fortran IV	Basic
Originally implemented on	IBM 360/67	Hewlett Packard, 2000C
Access to programs	Interactive/ batch	interactive
Provides card output	Yes	No
Software for students.		
Input variables	All conditions together	One condition at a time
Input variable order	Any order	Fixed-prompted order
Unspecified independent variables	Take default values	Take default values
Knowledge of commands	20 MESS commands	None required
Allows repeated measures	Yes	No
Provides summary statistics	Yes	Yes
Provides costs for each experiment	Yes	Yes
Software for instructors.		
Manipulation of hidden variables	Requires change of DATA file	Use codes with X variable
Ease of manipulation	Slow	Fast
Model building	By writing Fortran Sub-routine	By writing Basic program or use GMB
Time model building	Weeks	Days with GMB
Changing model initial-ization	Easy	More difficult
Change variable names	Easy	More difficult
Possible dependent variables	Up to 6	Up to 10
Possible independent variables	Up to 12 of each type	Up to 10 of each type
Possible total number of variables	50	24

FEAR SIMULATION

```
RUN
LESS

WHICH OF THE FOLLOWING MODELS DO YOU WISH TO RUN?
FEAR
OBESE
IMPRINT
SCHIZ
PARTY
TOT
?FEAR


EXPERIMENTAL PSYCHOLOGY
FEAR AND SEX SIMULATION

DO YOU WANT TO BE PROMPTED ON VARIABLE ORDER (Y OR N)?Y
PLEASE GIVE VALUE FOR NUMSUBJ ?15
MEASURE =?3
FEAR    =?2
TESTOST =?87


DO YOU WANT DETAILED OUTPUT?Y
  MOUSE      EXPLORE    THRUST
    1          82         12
    2          10          0
    3         104         34
    4          60          8
    5          77         20
    6         164         44
    7         106         48
    8          72         35
    9         106          9
   10          79         13
   11         145         29
   12         113          7
   13         110         17
   14         138         46
   15          95          6
EXPLORE MEAN = 97.4                S.D. = 37.4391
THRUST MEAN = 21.8667              S.D. = 16.1195
DO YOU WANT THE STANDARD ERROR OF THE MEAN DIFFERENCE (Y OR N)?N

DO YOU WANT TO BE PROMPTED ON VARIABLE ORDER (Y OR N)?
```

Figure 1 Student interaction with LESS (Fear simulation).

Figure 2 Student interaction with MESS

```
$SIGNON CCIO T=10
**LAST SIGNON WAS: 09:19.08
USER "CCIO" SIGNED ON AT 16:58.48 ON 04-23-73
```

The user signs on, specifying a machine time of ten seconds. Note that the computer does not print out the password card; this is a further aid in keeping your password secret.

```
SOURCE K2SH:IMPRINT
 ON 2A9A:IMPSIM 4=2A9A: IMPOATBIN 5=*MSOURCE*
EXECUTION BEGINS               6=*SINK* 7=-DATA 8=-PRINT
```

The computer prints out the users card (c) and the imprinting simulation begins.

```
MICHIGAN EXPERIMENTAL SIMULATION SUPERVISOR
       VERSION 3-S8   JANUARY, 1973
```

These lines give program title and authors' credits.

```
     IMPRINTING SIMULATION
     O. W. RAJECKI, FALL, 1970
     MODIFIED BY BOB STOUT, AUGUST, 1972

ENTER SUPERVISOR COMMAND
 >EXPT
```

Here the program asks for instructions. Card (d) is printed, which tells the program to begin an experiment.

```
ENTER EXPERIMENT ID LINE
A. EINSTEIN SECT. 029 TARG-ARO EXPT
```

The program asks for an identification line and card (e) is printed.

```
     JOB NO. 543176

     16:58.40    04-23-73


          UNIVERSITY OF MICHIGAN TERMINAL SYSTEM  (MODEL AR2Ø3)
```

```
    ENTER NO. OF EXPERIMENTAL CONDITIONS
(f) 4
```

This experiment is to have four conditions, as indicated on card (f).

```
    DEFINE EXPERIMENTAL CONDITION(S)
    TARG=CYL, HEN  ARO=3,5 WALK=mat
(g) @END
        4 CONDITION(S) DEFINED
(h)
```

The program asks for experimental condition definitions. Cards (g) and (h) are printed.

The program agrees that four conditions have been defined.

```
 THE FOLLOWING VARIABLE SETTINGS ARE CONSTANT
      REARING=SOCIAL    ACROSS ALL CONDITIONS:
         AGE=RANDOM          WALK=MATCHED
        TEST=1.000          INDUCT=MECH

   VARIABLE SETTINGS FOR CONDITION  A
       TARGET=CYLINDER
      AROUSAL= 3.000

   VARIABLE SETTINGS FOR CONDITION B
       TARGET=HEN
       AROUSAL= 3.000

   VARIABLE SETTINGS FOR CONDITION C
       TARGET=CYLINDER
      AROUSAL=5.000

   VARIABLE SETTINGS FOR CONDITION D
       TARGET=HEN
      AROUSAL= 5.000

(i) ENTER NO. OF SUBJECTS IN EACH GROUP
    15
```

The program states which variables are to remain constant across all the four conditions. Note that the variables--REARING, INDUCT, and AGE--have been set to their default values.

Each condition is defined. Note that the variable TARGET alternates first between CYLINDER and HEN; then the variable AROUSAL alternates between 3 and 5.

The programs asks for the number of subjects to be run in each condition, and then prints card (i).

```
A. EINSTEIN SECT. 029 TARG-ARO EXPT
16:59.08 APR 23, 1973
GROUP NUMBER 1
CONDITION(S): A
NUMBER OF SUBJECTS: 15

   TEST1   SCORES
        2.00     2.80     0.600     1.30     1.30
        0.500    0.700    3.70      1.70     2.50
        1.10     0.900    1.40      6.40     5.20

   NO. OF SS WITH COMPLETE DATA: 15

   VARIABLE: TEST1
   MEAN: 2.140
   VARIANCE: 3.031
   STD. DEVIATION: 1.741
```

The experimental simulation begins: the first line is the identification line, then comes the time and date of the experiment. Group Number 1 is to be run under condition A and has 15 subjects.

The score for each subject is printed out.

The scores are analyzed and the statistics are printed out.

```
A. EINSTEIN SECT. 029 TARG-ARO EXPT
16:59.09 APR 23, 1973
GROUP NUMBER 2
CONDITION(S): B
NUMBER OF SUBJECTS: 15

   TEST1   SCORES
        4.30     1.50     3.30      1.50     3.10
        0.700    1.90     4.00      4.00     0.200
        5.10     0.0      0.600     1.50     2.60

   NO. OF SS WITH COMPLETE DATA: 15

   VARIABLE: TEST1
   MEAN: 2.287
   VARIANCE: 2.613
   STD. DEVIATION: 1.616
```

The output for conditions B-D follows the same format as the output for condition A.

A. EINSTEIN SECT. 029 TARG-ARO EXPT
16:59.09 APR 23, 1973
GROUP NUMBER 3
CONDITION(S): C
NUMBER OF SUBJECTS: 15

TEST1 SCORES

6.90	12.6	7.60	1.20	9.00
10.7	7.50	1.20	7.50	9.50
6.50	1.90	0.900	1.30	10.3

NO. OF SS WITH COMPLETE DATA: 15

VARIABLE: TEST1
MEAN: 6.307
VARIANCE: 15.93
STD. DEVIATION: 3.992

A. EINSTEIN SECT. 029 TARG-ARO EXPT
16:59.09 APR 23, 1973
GROUP NUMBER 4
CONDITION(S): D
NUMBER OF SUBJECTS: 15

TEST1 SCORES

7.20	8.40	7.40	0.800	8.10
9.90	12.7	7.50	1.10	0.600
8.40	0.0	12.0	0.0	7.50

NO. OF SS WITH COMPLETE DATA: 15

VARIABLE: TEST1
MEAN: 6.107
VARIANCE: 19.43
STD. DEVIATION: 4.408

EXPERIMENT COMPLETED.

Notification of completion of the experiment. The user knows that all went well with this run.

(j) ENTER SUPERVISOR COMMAND
>STOP

NUMBER OF EXPERIMENTAL RUNS 1
NUMBER OF GROUPS SIMULATED 4

The program asks for another simulation supervisor command. Card (j) is printed and the program tallies the number of experiments and groups simulated.

$SOURCE PREVIOUS
(k) $SIGNOFF

The user signs off with card (k) and MTS prints out the signoff statistics.

```
**** ON AT 16:59.48      04-23-73
**** OFF AT 16:59.18     04-23-73
**** ELAPSED TIME           .483 MIN.
**** CPU TIME USED         4.441 SEC.           $.35
**** CPU STOR VMI          1.816 PAGE-MIN.      $.09
**** WAIT STOR VMI          .162 PAGE-HR.
**** CARDS READ           20                    $.02
**** LINES PRINTED       227                    $.10
**** PAGES PRINTED        11                    $.04
**** DRUM READS          114
**** APPROX. COST OF THIS RUN IS      $.58

**** DISK STORAGE         15     PAGE-HR.
**** APPROX. REMAINING BALANCE:      $4.61

**LAST SIGNON WAS:  09:19.08   04-23-73
```

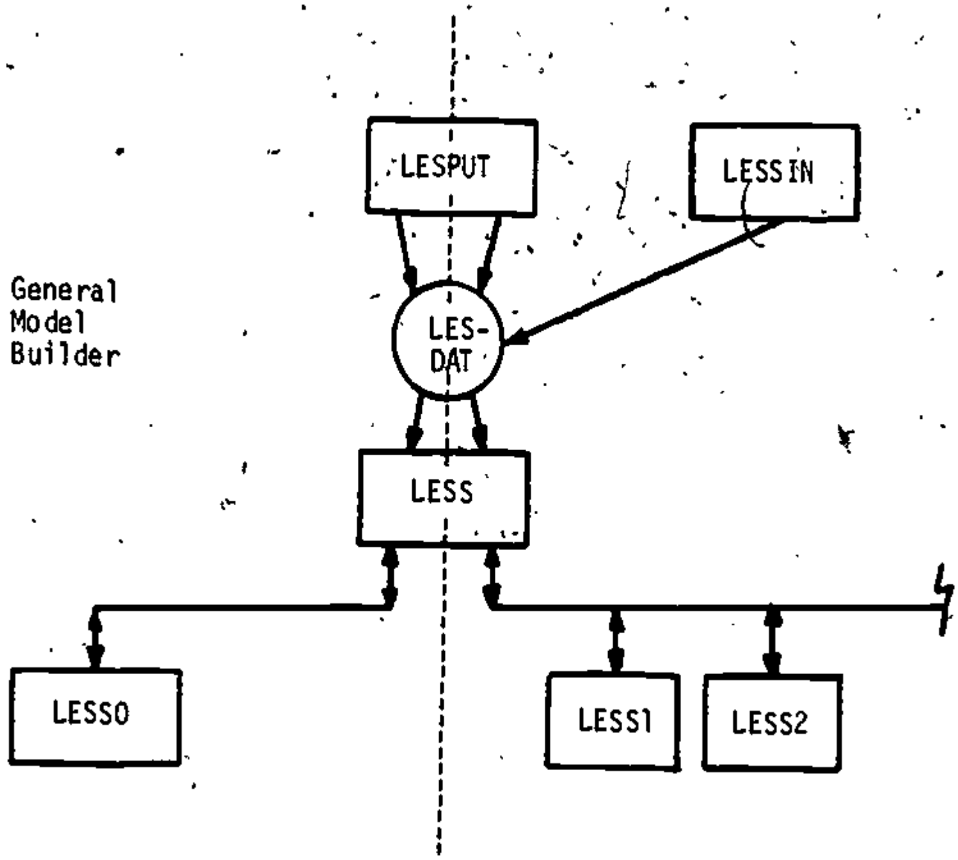

Figure 3 LESS system flowchart

GENERAL FLOWCHART FOR THE MESS SYSTEM

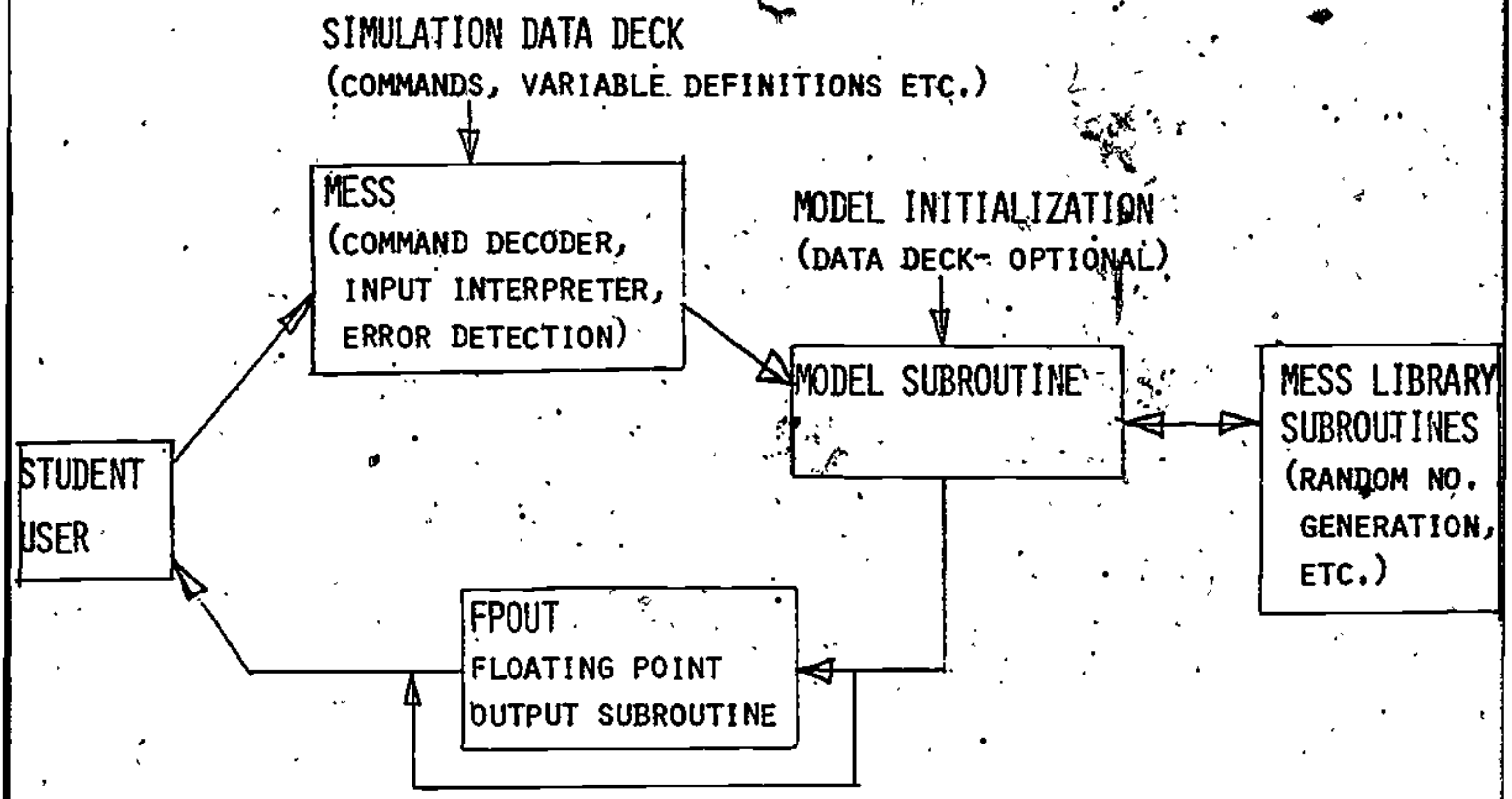

Figure 4 MESS system flowchart